Assignment 1

Data Processing

With Databricks

Shashikanth Senthil Kumar
Student ID: 25218722

2025 SPRING

94693 - Big Data Engineering
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

1. Introduction

This project focuses on analyzing the **New York City Taxi and Limousine Commission (TLC)** dataset using **Databricks Spark**. The dataset contains nearly a billion of yellow and green taxi trip records, including timestamps, pickup and drop-off locations, distances, fares, payment details, and passenger counts. The objective is to clean and prepare the dataset, answer business questions using SQL, and build predictive models to estimate trip fares.

2. Project Overview

The assignment is divided into three parts:

1. **Data Ingestion and Preparation** – Cleaning, joining, and storing the dataset in Delta format.
2. **Business Questions** – Using SQL to generate insights about trips, passengers, revenues, and taxi performance.
3. **Machine Learning** – Building and comparing models to predict the **total trip amount**.

This workflow highlights the **end-to-end big data engineering lifecycle**: ingest, transform, analyze, and model at scale.

3. Part 1 : Data Ingestion and Preparation

i. Data Acquisition

The raw datasets were obtained using the provided notebook, which downloads the **Yellow** and **Green** taxi trip records. Additionally, the **location reference CSV** was downloaded manually and uploaded to Databricks, stored in the same path as the taxi datasets for consistency.

ii. Environment Setup

We connected to a **Databricks serverless node** with the following configuration:

- **Memory:** 32 GB
- **Dependencies installed:** numpy, scikit-learn, threadpoolctl==3.2.0

This ensured the environment could handle large-scale processing and machine learning tasks efficiently.

iii. Data Cleaning and Validation

A systematic cleaning process was applied to ensure only realistic and valid trips remained in the dataset. The major steps were:

1. **Trips finishing before starting time** – Removed.

2. **Speed calculation and negative speed filtering**
    - Trip time was calculated as dropoff_datetime – pickup_datetime.
    - Speed (mph) was calculated using speed = distance / time.
    - Trips with **negative speeds** were removed.

3. **High speed filtering** – Removed trips exceeding NYC speed limits (threshold: **35 mph**, slightly above the legal limit of 30 mph).

4. **Trip duration filtering** – Removed trips shorter than **1 minute** or longer than **6 hours**.

5. **Trip distance filtering** – Removed trips shorter than **0.1 miles** or longer than **100 miles**.

6. **Payment type validation** – Only valid codes from the TLC data dictionary were accepted (0–6). No invalid values found.

7. **RatecodeID validation** – Only valid codes from the TLC data dictionary (1–6, 99) were accepted. No invalid values found.

8. **VendorID validation** – Only valid codes from the TLC data dictionary (1,2,6) were accepted. Invalid vendor IDs found and removed.

9. **Handling null values**

    - **Green Taxi:** Nulls found in store_and_fwd_flag, RatecodeID, passenger_count, ehail_fee, improvement_surcharge, payment_type, trip_type, and congestion_surcharge. Removing rows with null trip_type also eliminated most nulls, leaving only optional charges (ehail_fee, improvement_surcharge, congestion_surcharge).

    - **Yellow Taxi:** Nulls found in store_and_fwd_flag, RatecodeID, passenger_count, improvement_surcharge, congestion_surcharge, and airport_fee. Removing rows with null RatecodeID eliminated most nulls, leaving only optional charges.

10. **Passenger count validation** – Removed trips with **<1** passenger or **>6** passengers.

11. **Datetime validation** – Restricted trips to the official ranges:
    - Yellow taxis: **2009-01-01 to 2025-06-30**

    - Green taxis: **2013-01-01 to 2025-06-30**
      Trips outside these ranges were removed.

12. **Negative values** – Removed all records with negative values in monetary fields.
    - **Green Taxi:** Negative values in fare_amount, extra, mta_tax, tip_amount, tolls_amount, improvement_surcharge, total_amount, congestion_surcharge.
    - **Yellow Taxi:** Negative values in fare_amount, extra, mta_tax, tip_amount, tolls_amount, improvement_surcharge, total_amount, congestion_surcharge, airport_fee.

13. **Total amount threshold** – Removed trips with total fare **> $100**, as determined from dataset statistics (min, max, mean, median).

iv. Row Counts Before and After Cleaning

- **Original dataset size:** 991,467,464 rows
- **Final dataset size after cleaning:** 944,480,149 rows
- **Percentage removed:** ~4.7% (well below the 10% threshold)

v. Schema Standardization and Integration

Since Yellow and Green datasets had slightly different schemas, we standardized column names and added missing fields:

- **Green Taxi:**
    - lpep_pickup_datetime → pickup_datetime
    - lpep_dropoff_datetime → dropoff_datetime
    - Added airport_fee (null values)
    - Added taxi_colour column with constant value **"green"**

- **Yellow Taxi:**
    - tpep_pickup_datetime → pickup_datetime
    - tpep_dropoff_datetime → dropoff_datetime
    - Added ehail_fee and trip_type (null values)
    - Added taxi_colour column with constant value **"yellow"**

Finally, both datasets were combined using **UNION ALL** to form a unified taxi trip table.

vi. Location Data Integration

- A temporary view was created from the **location reference CSV**.

- The combined taxi dataset was **left-joined** with the location data on PULocationID and DOLocationID, enabling meaningful pickup/dropoff borough and zone analysis.

vii. Final Storage

The cleaned and enriched dataset was saved as a **Delta Table** in Databricks for efficient querying and machine learning.

viii. Final Row Validation

- Verified that less than 10% of data was removed during cleaning.
- Original dataset size: **991,467,464 rows**.
- Final dataset size: **944,480,149 rows**.
- This confirms that data cleaning preserved over 90% of the records.

4. Part 2 : Business Questions

Question 1: Monthly Trip Analysis

We grouped the dataset by **year and month** to explore taxi demand patterns. The screenshot shows monthly totals along with the busiest day, busiest hour, average passengers, and payment details.

	year_month	total_trips	top_day_of_week	top_hour_of_day	avg_passengers	avg_amount_per_trip	avg_amount_per_passenger
1	2009-01	962	Thursday	0	1.67	18.36	15.22
2	2011-01	2	Monday	23	1	12.8	12.8
3	2011-02	1	Tuesday	0	1	15.8	15.8
4	2014-01	13426273	Friday	19	1.7	13.93	11.25
5	2014-02	13155422	Saturday	19	1.69	14.15	11.48
6	2014-03	16008539	Saturday	19	1.68	14.25	11.58
7	2014-04	15452234	Wednesday	19	1.68	14.56	11.82
8	2014-05	15780871	Friday	19	1.68	15.09	12.23
9	2014-06	14768615	Sunday	19	1.68	15.13	12.26
10	2014-07	13994779	Thursday	19	1.68	14.75	11.95
11	2014-08	13545956	Friday	19	1.68	14.93	12.06
12	2014-09	14399478	Tuesday	19	1.66	15.17	12.36
13	2014-10	15453364	Friday	19	1.66	15.07	12.28
14	2014-11	14504801	Saturday	19	1.66	14.88	12.1
15	2014-12	14407912	Wednesday	19	1.66	14.96	12.14

138 rows | 1m 29s runtime Refreshed now

Insights from the results:

- **Trip Trends:** High volumes were observed during 2014–2019, followed by a sharp decline in 2020–2021 due to the COVID-19 pandemic. A gradual recovery appears from 2022 onwards.
- **Busiest Days & Hours:** Thursdays,Fridays and Saturdays are the busiest days in most years, with evening hours (around 6–7 PM) consistently recording peak demand.
- **Passenger & Payment Behavior:** Average passengers per trip remained stable (~1.4–1.7). The average fare per trip shows an upward trend across the years, reflecting inflation and changing travel behavior.

Question 2: Trip Characteristics by Taxi Colour

We compared **yellow and green taxis** on trip duration, distance, and speed. The screenshot summarizes the averages, medians, minimums, and maximums.

	taxi_colour	avg_duration_m	median_duration_m	min_duration_m	max_duration_m	avg_distance_km	median_distance_km
1	green	13.58	10.50	1.02	359.98	4.67	3.07
2	yellow	14.23	11.17	1.02	359.98	4.65	2.74

min_distance_km	max_distance_km	avg_speed_kmh	median_speed_kmh	min_speed_kmh	max_speed_kmh
0.18	160.24	20.15	18.49	0.03	56.33
0.18	160.45	18.44	16.49	0.03	56.33

Insights from the results:

- **Trip Duration:** Yellow taxis have slightly longer average trip times than green taxis.
- **Trip Distance:** Both taxi types cover similar average distances, but green taxis show a higher median, suggesting they serve slightly longer mid-range trips.
- **Trip Speed:** Green taxis maintain higher average speeds, while yellow taxis tend to move more slowly, likely due to denser traffic.

Question 3: Trips Summary by Taxi Colour, Boroughs, Time, and Fare

We analyzed yellow and green taxi trips across pickup and drop-off boroughs, months, days of the week, and hours. The table output summarizes total trips, average distance, average fare per trip, and total fare collected.(Attached first 15 rows of 131,643 rows).

	taxi_colour	pickup_...	dropoff_...	month	day_of...	hour	total_trips	avg_distanc...	avg_amount...	total_a...
1	green	Manhattan	Brooklyn	1	Sunday	0	122	21.12	47.02	5736.01
2	green	Unknown	Manhattan	1	Sunday	0	5	5.31	17.68	88.4
3	green	Unknown	Unknown	1	Sunday	0	15	5.73	15.9	238.52
4	green	Unknown	Bronx	1	Sunday	0	3	7.17	16.63	49.9
5	green	Brooklyn	Bronx	1	Sunday	0	39	23.12	47.72	1860.91
6	green	Manhattan	Manhattan	1	Sunday	0	7721	3.66	11.9	91878.93
7	green	Queens	Unknown	1	Sunday	0	29	18.78	41.91	1215.26
8	green	Bronx	Queens	1	Sunday	0	37	16.83	38.95	1441.01
9	green	Queens	Brooklyn	1	Sunday	0	764	10.49	25.16	19221.16
10	green	Queens	Manhattan	1	Sunday	0	1072	10.22	26.71	28638.14
11	green	Brooklyn	Queens	1	Sunday	0	1856	9.38	23.49	43597.33
12	green	Manhattan	Bronx	1	Sunday	0	1266	6.19	15.62	19769.88
13	green	Brooklyn	Unknown	1	Sunday	0	18	12.87	39.84	717.13
14	green	Staten Island	Staten Island	1	Sunday	0	1	1.58	7.56	7.56
15	green	Brooklyn	Manhattan	1	Sunday	0	4968	9.16	26.12	129740.18

Insights from the results:

- **Trip Volumes:** Yellow taxis consistently dominate overall trip counts , especially for Manhattan–Manhattan rides. Green taxis have fewer trips but capture demand in the outer boroughs (Brooklyn, Queens, Bronx).
- **Monthly Trends:** Trips peak during the summer months (June–August), while January and February have the lowest counts. December shows strong total revenue due to holiday travel.
- **Day of Week:** Fridays and Saturdays record the highest number of trips and revenue. Sundays see fewer rides but higher average trip distances, often linked to airport or inter-borough travel.
- **Hourly Trends:** Morning (7–10 AM) and evening (4–8 PM) are peak travel times, dominated by short Manhattan trips.

Question 4: Share of Total Revenue by Top 10 Pickup→Dropoff Borough Pairs in 2024

We analysed the share of total revenue contributed by the top 10 pickup→dropoff borough pairs for 2024. The screenshot summarises the revenue share percentages for these pairs.

	pickup_borough	dropoff_borough	revenue_share_percent
1	Manhattan	Manhattan	66.6
2	Queens	Manhattan	14.7
3	Manhattan	Queens	6.17
4	Manhattan	Brooklyn	3.27
5	Queens	Brooklyn	3.13
6	Queens	Queens	2.85
7	Manhattan	Bronx	0.52
8	Brooklyn	Brooklyn	0.51
9	Brooklyn	Manhattan	0.44
10	Queens	Bronx	0.35

Insights from the results:

- **Highest Revenue Share:** Trips within Manhattan (Manhattan → Manhattan) contribute the largest share, at **66.6%** of total revenue.
- **Strong Inter-Borough Demand:** Queens → Manhattan trips account for the second highest share (**14.7%**), followed by Manhattan → Queens (**6.17%**).
- **Other Notable Pairs:** Manhattan → Brooklyn (3.27%), Queens → Brooklyn (3.13%), and Queens → Queens (2.85%) also contribute meaningfully.
- **Concentration of Revenue:** The top two borough pairs alone contribute over **81% of total revenue in 2024**, indicating a high concentration of demand in specific corridors.

Question 5: Percentage of Trips Where Drivers Received Tips

We analysed the proportion of taxi trips where drivers received tips. The result is summarised in the screenshot.

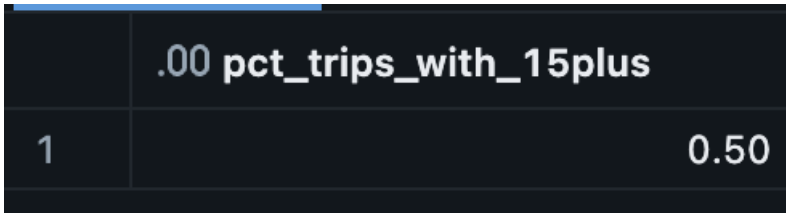
	.00 pct_trips_with_tips
1	63.19

Insights from the results:

- **Majority of Trips Include Tips:** Drivers received tips in **63.19%** of all trips.
- This indicates that tipping is a common practice among passengers, reflecting general passenger satisfaction or customary behaviour.
- It also suggests that for most trips, drivers earn additional revenue beyond the base fare.

Question 6: Percentage of Tipped Trips Where Tip Amount Was at Least $15

We analysed trips where drivers received tips to determine how often the tip amount was at least $15. The result is summarised in the screenshot.



	.00 pct_trips_with_15plus
1	0.50

Insights from the results:

- **Relatively Rare High Tips:** Only **0.50%** of trips where a tip was given had a tip amount of at least $15.
- This suggests that while tipping is common, **large tips are rare**, and most tips are smaller amounts.
- High tipping events likely correspond to longer trips or exceptional service, but they make up a very small proportion of overall tipped trips.

Question 7: Trip Duration Bins with Average Speed and Distance per Dollar

We classified all trips into bins based on trip duration and calculated the average speed (in km/h) and average distance per dollar (in km per $) for each bin. The results are summarised in the screenshot.

	duration_bin	avg_speed_kmh	avg_distance_per_dollar_km
1	Under 5 Mins	19.72	0.16
2	5–10 Mins	17.11	0.21
3	10–20 Mins	17.6	0.26
4	20–30 Mins	19.92	0.3
5	30–60 Mins	24.63	0.37
6	At least 60 Mins	21.78	0.61

Insights from the results:

- **Short Trips (<5 mins):** These trips have the lowest average speed (**19.72 km/h**) and the smallest distance per dollar (**0.16 km/$**), reflecting short, likely urban trips with frequent stops.
- **Moderate Trips (5–30 mins):** Average speeds range between **17.11 km/h** and **19.92 km/h**, and distance per dollar increases gradually from **0.21 km/$** to **0.30 km/$** as trip length increases.
- **Longer Trips (30–60 mins):** These trips have significantly higher average speed (**24.63 km/h**) and improved distance per dollar (**0.37 km/$**), indicating more efficient travel over longer distances.
- **Very Long Trips (≥60 mins):** These have slightly lower speed (**21.78 km/h**) than the 30–60 mins bin but the highest distance per dollar (**0.61 km/$**), suggesting long-distance trips offer greater value in distance covered per fare dollar.

Question 8: Duration Bin to Target to Maximise Driver Income

We analysed revenue efficiency across trip duration bins to determine which bin would maximise a driver's income. Efficiency was measured as the average revenue earned per minute of trip time. The results are summarised in the screenshot.

	duration_bin	total_trips	avg_amount_per_trip	avg_revenue_per_min
1	Under 5 Mins	135879590	7.25	2.17
2	5–10 Mins	279394833	10.24	1.4
3	30–60 Mins	67547050	45.83	1.17
4	10–20 Mins	337286070	16.06	1.14
5	20–30 Mins	116875252	26.33	1.09
6	At least 60 Mins	7497354	65.16	0.91

Insights from the results:

- **Short Trips (<5 mins):** These trips have the highest revenue efficiency (**2.17 $/min**) despite having the lowest average fare per trip (**$7.25**). This indicates that short trips allow for high trip turnover and maximised income per working hour.
- **Moderate Trips (5–10 mins):** These trips offer good revenue efficiency (**1.40 $/min**) with a higher average fare per trip (**$10.24**), making them a strong option for drivers when very short trips aren't available.
- **Longer Trips (10–60 mins):** Revenue efficiency gradually decreases from **1.17 $/min** to **1.09 $/min**, though the fare per trip increases significantly, reaching **$45.83** for 30–60 minute trips. These trips can still be worthwhile when time and passenger demand align.
- **Very Long Trips (≥60 mins):** These trips have the lowest revenue efficiency (**0.91 $/min**) despite having the highest fare per trip (**$65.16**), as the time required per trip limits overall hourly earnings.

5. Part 3 : Machine Learning

Baseline

As a starting point, we established a **baseline model** using the aggregated average fares by categorical and temporal features, consistent with the approach in Part 2, Question 3(c). The dataset was transformed to match the required baseline format, and RMSE values were computed for both training and test sets.

- **Training RMSE:** 8.26
- **Test RMSE:** 10.21

This provides a benchmark against which the performance of subsequent models can be evaluated.

Modelling Approach

The modelling task involves predicting taxi fares for the final three months of 2024 (October, November, and December).

Data Scope

Rather than using the full historical dataset (spanning 2009–2024), we restricted the training data to **2024 only** (January–September). This choice was driven by several factors:

1. **Economic shifts:** Older fare data would be less representative due to inflation and evolving fare structures.
2. **Changing patterns:** Seasonal, regulatory, and behavioural changes introduce year-to-year variability that could reduce model accuracy.
3. **Computational constraints:** The full dataset consists of billions of rows, making it computationally infeasible for model training without extensive resources.

Accordingly, the data was split as follows:

- **Train Set:** 2024 (January–September)
- **Test Set:** 2024 (October–December)

The resulting partitions were:

- Train: **25,689,161 rows × 21 columns**
- Test: **9,518,622 rows × 21 columns**

Then the dataset is Preprocessed using StandardScaler for Standardization and SimpleImputer for Missing value imputation.

Given the dataset's scale, running conventional models directly caused out-of-memory errors. To overcome this, we explored two complementary approaches.

Approach 1: Incremental Learning with Full Data

In this approach, we leveraged **incremental learning algorithms** capable of handling large datasets via batch processing.

- **Training Strategy:**
  - Batch size: 200,000
  - Models trained using partial_fit on successive data batches.

- **Results (Test Set):**
  - **SGDRegressor:** RMSE = 73.64
  - **PassiveAggressiveRegressor:** RMSE = 72.73

While scalable, this approach struggled to capture complex non-linear relationships, resulting in relatively high prediction errors.

Approach 2: Stratified Sampling and Tree-Based Models

To balance representativeness and efficiency, we designed a **stratified sampling procedure**.

- **Method:**

  We applied stratified sampling to preserve the distribution of key categorical features while reducing data size.

  - **Composite Strata Creation:** A temporary strata column was formed by concatenating features such as pickup_borough, taxi_colour, payment_type etc.
  - **Fractions Dictionary:** Each unique stratum was assigned a uniform sampling fraction of **0.04**, stored in a Python dictionary.
  - **Sampling Execution:** PySpark's sampleBy() was used with the strata column and fractions dictionary, ensuring reproducibility with a fixed seed (**42**).
  - **Cleanup:** The temporary column was dropped.

- **Sample Size: 1,027,780 rows × 21 columns**

This stratified sample preserved the distributional characteristics of the original dataset while reducing computational complexity.

- **Models Evaluated:**
    - Random Forest Regressor
    - Decision Tree Regressor
    - Extra Trees Regressor
    - Gradient Boosting Regressor
- **Results (Test Set):**
    - **Random Forest:** RMSE = 1.66
    - **Decision Tree:** RMSE = 2.33
    - **Extra Trees:** RMSE = 2.35
    - **Gradient Boosting:** RMSE = 2.16

Comparison and Insights

- The **baseline model** (RMSE ≈ 10.21) provided a reasonable starting point.
- **Approach 1** demonstrated scalability but poor accuracy (RMSE > 70).
- **Approach 2** substantially outperformed both the baseline and incremental models, with **Random Forest achieving the best performance (RMSE = 1.66)**.

This confirms that **tree-based ensemble methods trained on a stratified sample** are highly effective for this problem, offering both computational feasibility and superior predictive accuracy.

6. Issues Faced and Solutions

Large Dataset Size

- *Issue:* Nearly a billion rows made processing and modelling computationally heavy.
- *Solution:* Used **Databricks Spark** for distributed processing, stored data in **Delta format**, and applied **stratified sampling** and **batch incremental learning** to reduce computational load.

Slow Queries

- *Issue:* Large-scale queries were slow.
- *Solution:* Stored datasets in **Delta format**, created **temporary views**, and optimised queries with pre-filtering.

Modelling Limitations

- *Issue:* Full dataset training caused memory issues.
- *Solution:* Used **stratified sampling** for a representative subset and trained **tree-based models**, with Random Forest achieving the best performance (RMSE = 1.66).

7. Conclusion

This project successfully demonstrated the end-to-end process of analysing a large-scale real-world dataset using Databricks Spark, combining big data engineering, SQL-based analytics, and machine learning. Through systematic data cleaning, schema standardisation, and integration, over 944 million taxi trip records were prepared for analysis without losing significant data quality.

The business analysis revealed valuable insights into trip patterns, demand trends, driver behaviour, and revenue efficiency. Notably, short trips (<5 mins) offer the highest revenue efficiency, while longer trips provide better distance per dollar but lower earnings per unit time. These insights can inform taxi fleet management, driver strategies, and fare optimisation.

On the modelling front, stratified sampling coupled with tree-based ensemble methods proved highly effective, achieving strong predictive performance (Random Forest RMSE = 1.66) while overcoming the computational challenges of processing nearly a billion records.

Overall, the project highlights the power of combining big data tools, robust data preparation, and advanced modelling to derive actionable insights from complex datasets. Future work could explore deeper feature engineering, real-time analytics, and deployment of predictive models for dynamic fare estimation and operational optimisation.

8. References

Databricks. (n.d.). *Getting started with Databricks*. Retrieved September 2025, from https://docs.databricks.com/

Apache Software Foundation. (n.d.). *PySpark documentation*. Retrieved September 2025, from https://spark.apache.org/docs/latest/api/python/index.html

Scribbr. (n.d.). *Stratified sampling: Definition, guide & examples*. Retrieved September 28, 2025, from https://www.scribbr.com/methodology/stratified-sampling/

New York City Taxi & Limousine Commission (TLC) Trip Record Data, August 2025, from https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page